Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

A Flexible Laboratory Environment Supporting Honeypot Deployment for Teaching Real-World Cybersecurity Skills

NEIL ELIOT[1], DAVID KENDALL[1], AND MICHAEL BROCKWAY[1]
[1]Northumbria University, Department of Computing and Information Sciences, Newcastle upon Tyne, NE1 8ST
Corresponding author: Neil Eliot (e-mail: neil.eliot@northumbria.ac.uk).

ABSTRACT In the practical study of cybersecurity, students benefit greatly from having full control of physical equipment and services. However, this presents far too great a risk to security to be permitted on university campus networks. This paper describes an approach, used successfully at Northumbria University, in which students have control of an off-campus network laboratory, with a dedicated connection to the Internet. The laboratory is flexible enough to allow the teaching of general purpose networking and operating systems courses, while also supporting the teaching of cybersecurity through the safe integration of honeypot devices. In addition, the paper gives an analysis of honeypot architectures and presents two in detail. One of these offers students the opportunity to study cybersecurity attacks and defences at very low cost. It has been developed as a stand-alone device that also can be integrated safely into the laboratory environment for the study of more complex scenarios. The main contributions of this paper are the design and implementation of: an off-campus, physical network laboratory; a small, low-cost, configurable platform for use as a "light-weight" honeypot; and a laboratory-based, multi-user honeypot for large-scale, concurrent, cybersecurity experiments. The paper outlines how the laboratory environment has been successfully deployed within a university setting to support the teaching and learning of cybersecurity. It highlights the type of experiments and projects that have been supported and can be supported in the future.

INDEX TERMS Cybersecurity, Network Security, Honeypot, Teaching

I. INTRODUCTION

When teaching cybersecurity in an academic environment, there are many considerations that need to be taken into account, not least respect for users' privacy and their access to learning and teaching facilities. Cybersecurity experiments can result in traffic or services being deployed into the network of the teaching environment in ways that may affect other users, for example, through service depletion, traffic redirection [1], [2], or unauthorised data capture. In the most serious cases, the legal rights of other users may be violated. For these reasons, there are restrictions on the network activities that are permitted on a typical university campus network, designed for general student access. For example, port security may be employed to prevent the attachment of unauthorised equipment to the network, or firewalls may be used to restrict outgoing / incoming traffic from the Internet. Quite rightly, students are not allowed administrative privileges to configure the computers and other devices attached to the main campus network.

It is clear that these restrictions severely limit the kind of work that can be undertaken in teaching real-world, practical cybersecurity skills, and the access to potentially useful teaching resources [1], [3] that can be made available to students. They also prevent the control over the network that is required for the effects of attacks to be isolated and analysed thoroughly. For example, in order to analyse the effects of an attack on the throughput of a protocol, it may be necessary to restrict other network activity, such as file transfers or service advertising that may impact on the experimental environment and invalidate the results. This is not possible on a main campus network.

These issues highlight the need for a clear specification of the requirements of a specialist network environment for the study of cybersecurity. This is one of the key topics of this

paper.

Another key topic is the role that can be played by honeypots in providing a stimulating environment for students to study practical aspects of cybersecurity. Honeypots are computer systems designed specifically for cybersecurity investigations [4], [5]. They exist in a variety of forms, e.g. as a "lab-in-a-box" (LiaB) [6]; as a full network deployment, incorporating switches, routers, and servers; as a simple program running on a computer, or as a virtualised platform. We classify honeypots in terms of the level of interaction and analysis that they offer, and also in terms of the volume of concurrent requests that they are able to process.

- ***Low Interaction:*** Implements coarse-grained services and captures only a low-level of detail about their implementation and their interaction with users. This type of honeypot can also be used to act as an attractor for bot-based attacks [7].
- ***High Interaction:*** Implements fine-grained services and captures a high-level of detail about their implementation and their interaction with users. This type of honeypot is called a "research" honeypot by Mairh et al. [8]. High-interaction honeypots can also be used to distract potential hackers from a genuine system (a decoy) [9], [10].
- ***Low Volume:*** Capable of supporting only a small number of concurrent requests. This type honeypot is often deployed in a research or teaching laboratory to investigate the details of a single attack type.
- ***High Volume:*** Capable of supporting a large number of concurrent requests. This type of honeypot is often deployed in an environment that results in the honeypot being subjected to high levels of usage from multiple sources.

These two categories provide four different types of honeypot based upon interaction and volume. We use the obvious acronyms to refer to each different type, as shown in Table 1.

TABLE 1. Honeypot types by interaction and volume

	Low Volume	High Volume
Low Interaction	LILV	LIHV
High Interaction	HILV	HIHV

This paper focuses on **L**ow-**I**nteraction-**H**igh-**V**olume (LIHV) and **H**igh-**I**nteraction-**L**ow-**V**olume (HILV) honeypots. The other two types of honeypot are less interesting, for different reasons. On the one hand, a **L**ow-**I**nteraction-**L**ow-**V**olume (LILV) honeypot has limited utility. If required, its functionality can be provided by an HILV honeypot, at no significant extra cost. For example, an exercise in installing and testing a tool such as Kippo [11] can be undertaken with an HILV honeypot just as easily as with an LILV honeypot. On the other hand, a **H**igh-**I**nteraction-**H**igh-**V**olume (HIHV) honeypot is expensive to deploy and time-consuming to reconfigure for different teaching scenarios. This type of honeypot is useful in a research environment, e.g. for use in long term Internet-based projects, but is not required for teaching a full range of practical cybersecurity skills at undergraduate level.

The main finding of this paper is that a stimulating environment for teaching practical, real-world cybersecurity skills, in a university setting, can be provided in a safe, cost-effective manner using a combination of a flexible, general purpose, network laboratory together with the sandboxed deployment of LIHV and HILV honeypots.

The paper is structured as follows: Section II discusses current research in approaches to teaching practical aspects of cybersecurity. Section III outlines the requirements of a flexible laboratory environment for teaching networking and cybersecurity. Section IV classifies honeypot architectures and discusses the two that we have found most useful for the teaching of networking and cybersecurity. Section V describes the design and implementation of a general purpose networking laboratory, suitable for the safe deployment of honeypots. Section VI discusses how the flexible laboratory environment has been used at Northumbria University for teaching networking and cybersecurity. Section VII concludes and proposes further work for the development and application of the laboratory for teaching and research.

II. RELATED WORK

There are many papers relating to the development and use of honeypot technologies. These papers cover both physical and virtual implementations, and their application to teaching cybersecurity.

Romney and Lanoy [2], Hiber et al. [12], Wannous and Nakano [13] and Marsa et al. [14] discuss the use of a variety of virtual environments for the deployment of laboratory/honeypot platforms and networking architectures. In our experience, virtualisation works well for network simulation but limits the range of practical cybersecurity activities that can be undertaken. The limitations extend to deployment, resource utilisation, and data capture. The main problems arise because the resources in a virtual environment are shared. For example, if multiple experiments are performed simultaneously in a laboratory based on VLAN technology, unexpected results may be observed due to cross boundary effects. We have seen a MAC flood experiment running on one VLAN cause unexpected packet loss in another experiment running at the same time on another VLAN. This can be caused by exhaustion of a resource shared by VLANs using the same switch, e.g. the switch's MAC table or backplane. This issue should cause concern to anyone considering performing cybersecurity experiments on a VLAN connected to their main university campus network.

Experiments in a virtual environment may breach a sandboxed virtual host, so impacting on other experiments. The physical networking components of a virtual server are often shared across the virtual hosts even though the hosts use virtualised addresses. Again, this can produce unexpected results when experiments are executed simultaneously. When

conducting remote attack experiments, the network capture capabilities are also complicated by virtualisation, due to the mixture of traffic from multiple experiments. For example, a simple reconnaissance scan of a network can compromise other experiments.

Although it is often possible to propose a solution to any particular problem caused by the sharing of resources, it is very difficult to be confident that one has considered all such problems that may arise when using a virtual environment. Having said that, we do often use virtualisation to add resources to our physical architecture, but only in circumstances where the sharing of the virtualised resources can be carefully controlled.

Abler et al. [15] describe a more independent, hardware-based environment for cybersecurity experiments, but their environment still relies on VLAN technologies and requires a large number of routers, switches and servers. Their environment is configurable but it seems that experiments involving attacks on network infrastructure could cause problems for multiple, simultaneous users.

Salah et al. [16] propose the use of cloud services such as Amazon Elastic Compute Cloud (EC2) for the deployment of configurable environments for cybersecurity. However many of the issues relating to virtualisation are reflected in this approach. In addition, the usage agreements for the EC2 platform places restrictions on what can be done in this environment. Section 6 of the Customer Agreement states that the service could be suspended if *"... your or an End User's use of the Service Offerings (i) poses a security risk to the Service Offerings or any third party, (ii) could adversely impact our systems, the Service Offerings or the systems or Content of any other AWS customer"* [17].

Lee et al. [18] discuss an approach based on a simple physical LAN, with multiple clients and servers located on a subnet. Their environment is very popular for cybersecurity competitions in which teams can compete in activities such as capture-the-flag or time trials. The environment provides limited access to network traffic, which compromises its effectiveness as a tool for the analysis of specific attacks, but it does provide users with a platform for trying out attacks, usually using well-established tool sets, such as those provided by Kali Linux [19]. This type of platform works well for events such as those organised by the Cyber Security Challenge UK organisation [20] where cybersecurity is treated as a competitive activity rather than a topic for detailed investigation.

The low cost of the HILV honeypots described in this paper enable us to extend the multi-honeypot experimental environments discussed by Duffany [22], since it becomes affordable to increase the number of available honeypots for distributed data capture. Our HILV design also produces a honeypot that is small enough to be portable, and therefore easily deployed in different locations, not only the laboratory but even a student's home environment.

In comparison with other approaches, we have found that the integration of HILV honeypots with an LIHV honeypot in a dedicated network laboratory gives a highly configurable, cost-effective, safe environment for running many different networking and cybersecurity projects and activities. For example, our environment is capable of investigating all of the attack techniques discussed by Spitner [21], as well as many others.

III. LABORATORY REQUIREMENTS

This section discusses the main requirements of the honeypot technologies and the general purpose network laboratory for the delivery of networking and cybersecurity modules. The teaching environment consists of three distinct components: the small scale HILV honeypots (*requirements 1,2,3,4,5*), the LIHV honeypot (*requirement 6*), and the general purpose networking laboratory (*requirements 7,8*).

A. HILV HONEYPOT

Requirement 1: The HILV honeypot must allow students: to deploy basic networking services on multiple low-cost servers (e.g. DHCP/DNS/HTTP/DB); to attack these services; and to capture and analyse the details of the network traffic and server events that are generated.

Requirement 2: In order to satisfy Requirement 1, while protecting the rest of the laboratory from the effects of attacks, there needs to be a mechanism that allows students to connect their servers to the general networking laboratory via port mapping or address forwarding.

Requirement 3: The servers need to have limited hardware resources in order to allow the analysis of resource exhaustion attacks without requiring a large number of attacking end points, e.g. it should be possible to launch a successful resource exhaustion attack from a botnet consisting of a few machines rather than hundreds or thousands.

Requirement 4: The honeypot must provide scope for additional services and devices to be added. It must also provide the ability to cascade multiple honeypots to create a honeynet [23]–[25].

Requirement 5: The honeypot must facilitate effective network traffic capture to ensure the integrity of any network analysis e.g. identifying network transactions in a website defacement or denial of service (DoS or DDoS) attack, or spoofed packets in ARP poisoning for man-in-the-middle (MitM) attacks [26], [27].

B. LIHV HONEYPOT

Requirement 6: There needs to be a facility that allows large numbers of students to test existing cybersecurity tools and to provide them with the ability to develop and test their own tools.

C. GENERAL NETWORKING LABORATORY

Requirement 7: There must be a flexible, reconfigurable, base laboratory environment, isolated from the main university campus network, for students to carry out their normal studies of networks, operating systems and network services. Students require administrative access to the basic

networking equipment such as routers, switches, and desktop machines for installation and configuration of general purpose tools and virtual environments. These activities would normally be prohibited on a university campus network [14]. This is the case at Northumbria University, although limited access is provided to staff rooms via campus VLAN connections. However, these connections are to be phased out in 2018. There also needs to be a facility to deploy a standard software environment to all devices with minimal administrative intervention.

Requirement 8: The laboratory network should implement a security policy, independent of the standard university policy, controlling access to the Internet. This is to allow students to access security sites and relevant software packages. Many academic networks block access to cybersecurity tool sites from their specialist and general access laboratories [1], [3], as well as from the open access areas used by students. Tools such a Metasploit [28] or the Kali Linux distribution [19] are usually blocked, as are cybersecurity information sites such as http://www.hak5.org or https://www.exploit-db.com/. It must be possible to lift these restrictions in a laboratory supporting cybersecurity studies.

IV. HONEYPOT ARCHITECTURES

As shown in Table 1, we consider four types of honeypot architecture.

- ***LILV*** This type of honeypot is used when a basic testing platform is required to identify how a service is being attacked but no other interactions need to be investigated. For example, a software system, such as Kippo [11], can be studied to discover how it logs transactions to a file or a database.
- ***LIHV*** This type of honeypot is used when carrying out analysis of a high volume of service requests while capturing only a limited amount of service and system activity data, e.g. when analysing multiple concurrent authentication attacks on an FTP server by investigating only the service log files. This type of honeypot can also be used in a general purpose networking laboratory with students to allow them to investigate authentication tools such as Hydra and xHydra [29] and to be a target when they are developing their own tools such as an authentication-based botnet.
- ***HILV*** This type of honeypot is used when there is a low volume of service requests but fine-grained data capture is required. For example, when investigating a DNS enumeration attack, a single AXFR [30] query may be the only service request required to initiate the attack, but the effect of this query, and the high-level of activity it generates, can be captured in great detail. A similar example is the profiling of a Wordpress [31] site using WPScan [32].
- ***HIHV*** This type of honeypot is used when a system is being thoroughly tested with a high volume of service requests (pressure testing) and involves large numbers of high-powered servers capable of supporting high levels of concurrent interactions. These honeypots provide detailed data capture capabilities of all the services and the inter-service interactions as well as system transactions e.g. SQL queries and responses. They are usually deployed as "real" systems for penetration testing and are often exposed to the Internet to analyse the effects of unsolicited attacks.

The two types of honeypot deployed in the laboratory are HILV and LIHV. These are now presented in more detail.

A. HILV HONEYPOT

The HILV honeypot architecture provides an isolated environment that can be connected to the main laboratory infrastructure in a controlled way using a commercially available cable router, as shown in Fig. 1.

Figures 2 and 3 shows the complete device as deployed in the specialist teaching laboratories. The complete HILV honeypot consists of:

- A router for traffic management to and from the laboratory infrastructure via NAT and address forwarding. (connected via the green cable).
- 4 Raspberry Pi boards [33] for service deployment.
- A managed switch for packet capture:
  - 1 port (port 8) setup as a monitor usually connected to a PC running TCPDump or Wireshark (red cable).
  - All other ports are mirrored to the monitor (ports 1–7).
  - 1 port (port 7) links the switch and router.
  - 4 ports (ports 1–4) are connected to the 4 Raspberry Pi servers.
  - 2 spare ports (ports 5,6) are available for additional services, clients, or devices to be added (blue cables).

The two additional devices shown in Fig. 1 could be PCs acting as attack entry points or victims. These devices could also be PCs supporting virtualisation to extend the honeypot's functionality.

The use of Raspberry Pi boards as the main servers for the honeypot simplifies reconfiguration and rebuilding of the architecture. The operating system (and configured services) are stored on removable media (micro SD cards) which simplifies server recovery and the maintenance of multiple configurations. Images of the honeypot's base configuration can easily be restored. Multiple configurations can be kept on different image sets and students can keep individual projects as a set of micro SD cards that they retain for the duration of a project.

The approximate cost of the basic HILV honeypot, without additional devices or the monitoring PC attached, is ≈£300 a breakdown of the costs is shown in Table 2. This offers a highly stimulating environment for the teaching of practical cybersecurity skills at a very low cost.

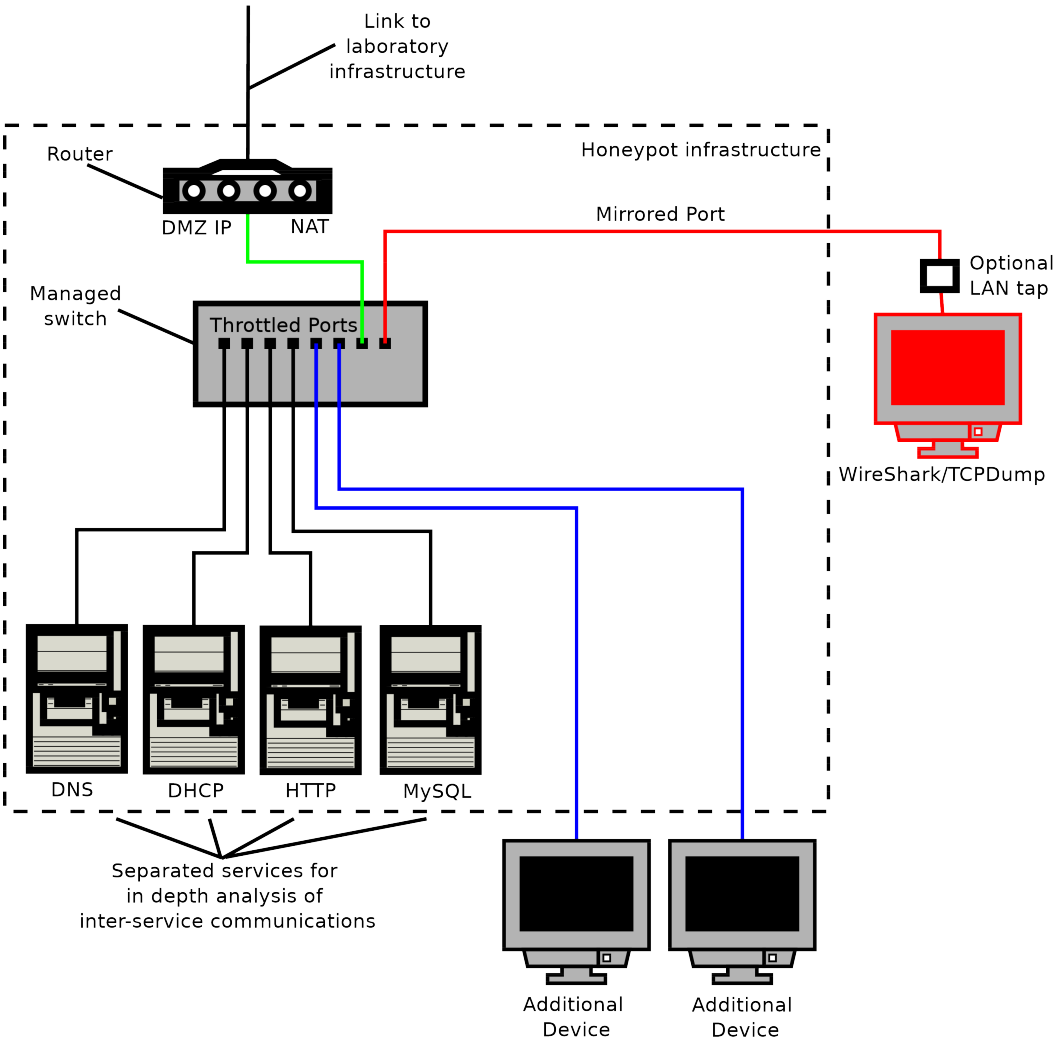

FIGURE 1. HILV honeypot overview

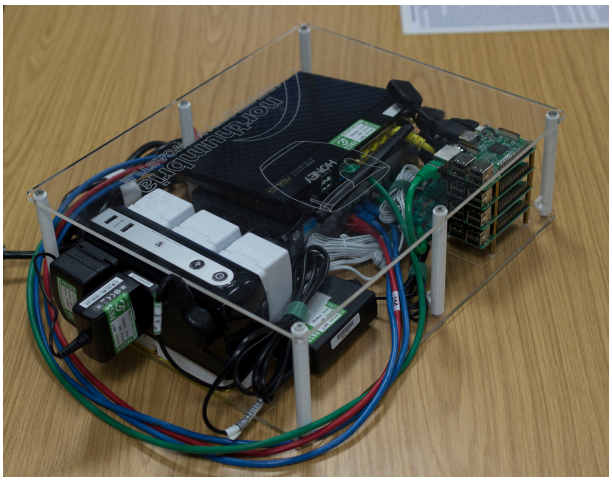

FIGURE 2. HILV honeypot side view

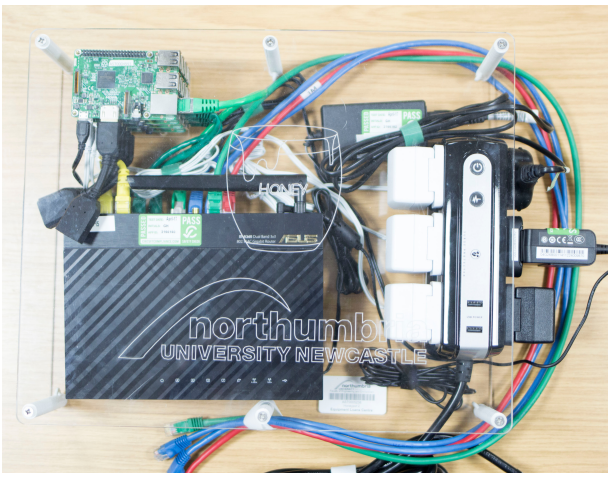

FIGURE 3. HILV honeypot top view

TABLE 2. HILV Costs

Product	Quantity	Unit Cost	Total
TP Link TL-SG108E 8 port switch	1	£29.99	£29.99
Raspberry Pi 3	4	£33.00	£132.00
Raspberry Pi PSU	4	£8.89	£35.56
TP Link TL-R470T+ Cable Router	1	£34.99	£34.99
Ethernet cables 1m (3 Pack)	1	£15.31	£15.31
Ethernet cables 0.25m (5 pack)	1	£8.56	£8.56
Masterplug 6 socket extension	1	£16.99	£16.99
Perspex Fabrication	1	£30.00	£30.00
			£303.40

Address Range Support: The HILV honeypot environment must reside on a different subnet from the main laboratory network in order for the double NAT'd configuration to function correctly. This is achieved by assigning a private class A IP address block, providing more than 16 million addresses.

Router Configuration: The HILV honeypot's ADSL router is configured to provide only routing services by disabling all other services, e.g. NAS and DHCP facilities. This configuration allows all infrastructure protocols to be managed from within the honeypot using small-scale servers (Raspberry Pi boards). Configuring the services on separate servers allows analysis of inter-service activity during normal network activity and during an attack.

A low-cost router such as a TP Link TL-R470T+ [34] is adequate for use in the HILV honeypot. The router is configured to use the laboratory-based DHCP server to acquire an address. The IP address is allocated from a reservation which allows a consistent mapping of the forwarded IP address from the Internet-based router to the HILV honeypot router. Defining the mapping in this way allows external host names (URLs) to be mapped to the honeypot for Internet-based attack analysis. Forwarding of traffic from the router into the honeypot can be enabled and disabled when required.

For direct access to the honeypot from the laboratory, IP forwarding is configured to "point" to a target machine within the honeypot, as shown in Fig. 4. In small-scale routers, this operation is normally referred to as a DMZ (demilitarised zone) redirection [35].

Switch Configuration: A key component of the HILV honeypot is a commercially available, 8-port, managed switch. The switch supports layer 2 management [36] to provide two specific technologies: port mirroring and port throttling. Suitable switches include HP 2530-08 [37], TP-LINK TL-SG2008 [38], and TP Link TL-SG108E [39]. The combination of port mirroring and port throttling provides a reliable architecture for packet capture.

Port mirroring: Switches maintain an in-memory table of ports and MAC addresses to support the efficient delivery of frames. This ensures that packets are transferred port to port rather than being broadcast (port to port communications are known as "virtual circuits"). This technology complicates the process of packet capture. When using a honeypot, packet capture is a vital part of the architecture for the analysis of any network-based attack vector. Using a managed switch (as discussed above), it is possible to configure the ports on the switch to be mirrored to a specific port, as shown in Figs. 1 and 5. This allows all the network activity to be captured and analysed (*Requirement 8*). Switches can implement mirroring in one of two ways. Firstly, a monitor-only port can be provided, where the transmission capability of the port is removed, (as in the case of an HP 1810-G). This type of configuration prevents the monitoring device from adding traffic to the network. Alternatively, some manufacturers configure the mirroring port to offer full transmit/receive functionality, in addition to the mirroring capability (as in the case of a TP-SG108E). This allows the attached monitor to also be used as a device within the honeypot. If the monitor port provides full functionality, the addition of a LAN tap [40] can remove the transmission facilities of the port, creating monitor-only functionality when required, as shown in Fig. 1.

Bandwidth throttling: A further issue with the capture of the honeypot network traffic is the possibility of frame loss on the port to which the mirrored traffic is forwarded. Switches are designed to provide maximum transfer speed between ports. This is achieved through the switch's backplane. If the throughput of traffic on the mirrored ports exceeds the bandwidth of the port to which the traffic is mirrored, then frame loss at this port is inevitable. To prevent this occurring, the switch must be configured to throttle the throughput on the mirrored ports so as not to overwhelm the backplane. Figure 5 shows the basic configuration of a throttled environment. The port which has the frames forwarded to it must be configured to run at a speed that exceeds the total bandwidth of the mirrored ports. The effect of this is that as frames are transferred between ports, they are reliably replicated via the backplane to the monitor port. This configuration satisfies requirement 8.

Internet Support: Access to the Internet from the honeypot is possible through the double NAT'd configuration. This configuration provides isolation from the laboratory and the Internet. External response to a service request from a device in the honeypot is achieved through packet forwarding, as shown in Fig. 4. For this technique to function correctly the laboratory network and honeypot network must be on different subnets.

The Internet facility also supports IP address forwarding to allow Internet-based access into the honeypot. From the Internet, packets are forwarded to the address of a honeypot router which, in turn, forwards traffic to a target machine inside the honeypot, as shown in Fig. 4.

This configuration allows specific configurations to be exposed in order to capture Internet-based attacks and to support remote access to the honeypot for remote configuration

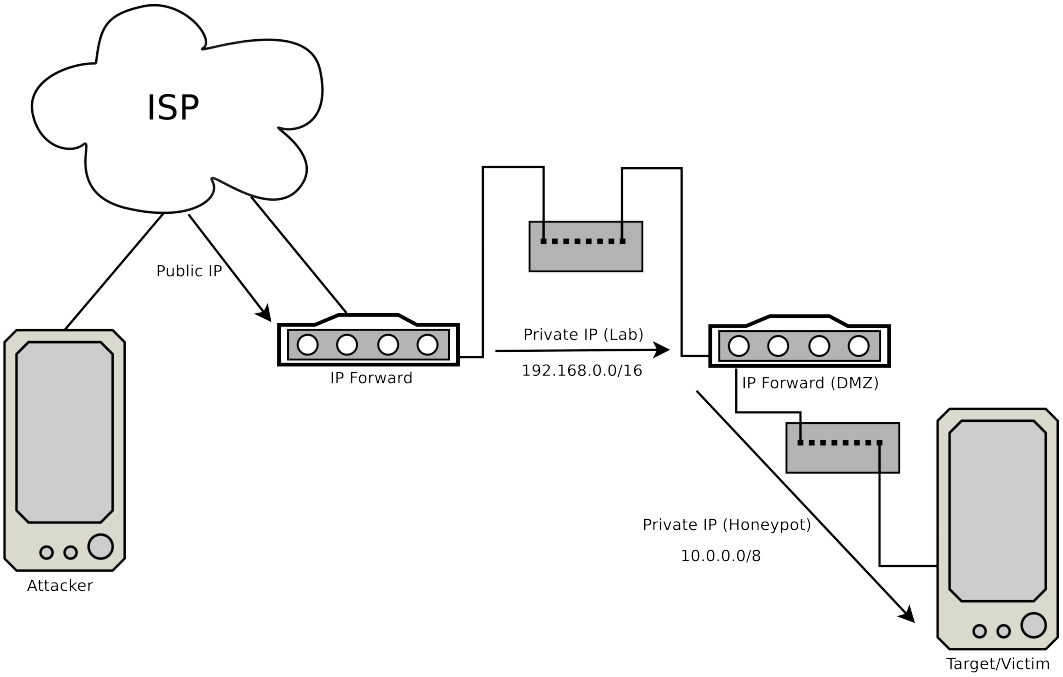

FIGURE 4. Address/Port forwarding

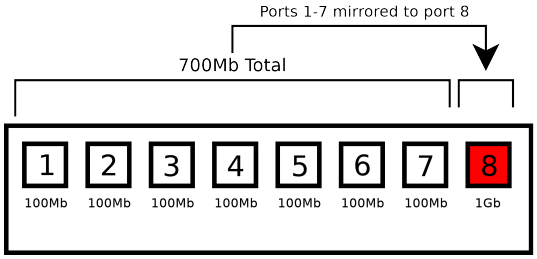

FIGURE 5. Port throttling

and monitoring.

B. LIHV HONEYPOT

The LIHV honeypot is not intended for use in a reconfigurable environment and does not require any monitoring or control of the network. No port monitoring or throttling is required and all activity monitoring is achieved using log files. The log files are made available for analysis of service-based access activity via a web-link.

The LIHV honeypot is a single 1U LAMP server (HPE ProLiant DL360 Gen9 Server [41] costing approx £2000), located in a separate cabinet, shown in Fig. 6. The cabinet is secured to prevent students having direct access to the hardware. It supports the general honeypot service to provide students with a platform to carry out simple authentication attacks using tools such as Hydra from within the Kali Linux [19] tool set. This honeypot also provides a target for the development of bespoke authentication attack tools in the final year undergraduate cybersecurity modules.

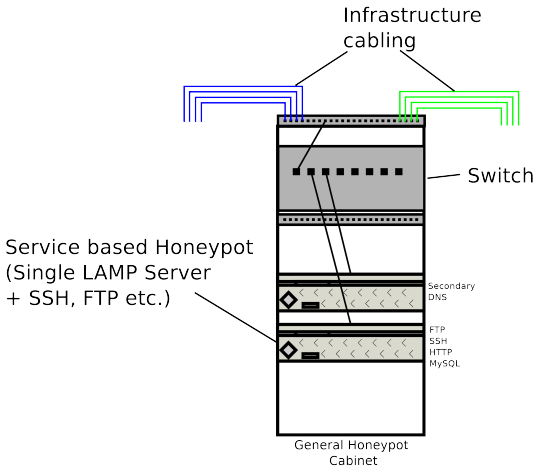

FIGURE 6. LIHV honeypot overview

This server is accessed directly in the laboratory, or via the Internet, as a bastion server [42]. This allows access from within the laboratory, but also from outside the laboratory to support directed learning tasks and also to support collaborative ventures.

V. GENERAL NETWORKING LABORATORY

A. OVERVIEW

The flexible laboratory environment comprises a secured communications cabinet, several open-access, teaching cabinets, and benches with desktop PCs for student use. The physical networking layer between devices is reconfigurable.

This is achieved by the use of a structured cabling architecture (Fig. 7). Internet access is provided by cascading each cabinet's "internet switch" to the central communication cabinet's "internet switch". A logical view of the internet facilities is shown in Fig. 8.

The communications cabinet is secured; students are not allowed access to it, in order to prevent them from reconfiguring the base network and the network services equipment. Figure 9 shows the main infrastructure cabling within the communications cabinet. The cabinet also contains the primary DHCP and DNS servers, a 1U intranet server, a database server and a NAS drive as shown in Fig. 10. It also includes a VDSL fibre link from the ISP. In the case of Northumbria University, the provider is BT Business [43]. A Draytek Vigor 2850 [44] router is used to manage the ISP connection.

Students have unrestricted access to the other cabinets, each of which is linked to a laboratory bench supporting 8–10 students, These cabinets contain a selection of switches, routers, IP telephony, IDS, and firewall equipment that is required for general computer networking modules, as shown in Fig. 11. They are linked back to the communications cabinet to provide internet access for the benches. Access to the internet from the desktop is achieved through a NAT connection from the ISP router in the communications cabinet. The structured cabling allows the resources of each cabinet within the laboratory to be linked to the desktop through structured cabling ports. Cross-cabinet connections ensure specialist resources are flexibly available to all benches e.g. switches, routers, firewalls, and IDS (Intrusion Detection Systems) etc.

B. ADDRESS RANGE SUPPORT

The use of virtualisation at the desktop (GNS3 [45], VMWare [46] and VirtualBox [47]), to support operating systems and networking modules, requires enough host addresses to support a large number of students deploying statically addressed virtual hosts. The base laboratory network infrastructure is configured as a single class B subnet. This provides 65,534 addresses, which is more than enough for each student to be allocated a block of contiguous IP addresses for each module, allowing multiple modules to be delivered concurrently without having IP address conflicts. This approach also allows equipment to be moved around in the laboratory without requiring address reconfiguration.

C. NETWORK SERVICE DEPLOYMENT

To support the study of operating systems and other general computing subjects, the network requires, as a minimum, DNS [48] and DHCP [49] services. These services are deployed across three small-scale, 1U servers (HPE ProLiant DL360 Gen9 Server [41]). Two servers are located in the communications cabinet as shown in Fig. 7. One server supports DHCP and DNS, which are integrated to create a DDNS [50] environment. The second server acts as the secondary DNS server. The third server is located in the LIHV honeypot cabinet and acts as a further secondary DNS server. The communications cabinet also contains a 1U server that provides the laboratory intranet (LAMP based) service, along with general purpose MySQL database services for the intranet and module content delivery.

Deployment of known desktop equipment is coordinated by creating reservation entries in the DHCP service. The DHCP server then automatically creates the forward and reverse DNS entries in the DNS architecture as the equipment boots. Students can also connect their own devices which are allocated a network configuration from a DHCP address pool.

D. TEACHING FACILITIES SUPPORT

Teaching of standard technologies, such as switching and routing, use the physical equipment within the cabinets. The laboratory also supports network virtualisation through GNS3, which can be integrated with the physical equipment when necessary. The teaching of the OS-based technologies is supported through the use of desktop virtualisation, allowing each student to have multiple clients and servers running simultaneously on a single machine. For large-scale scenarios, which are required on some modules, the virtual machines may be deployed across several desktop machines. These large scale deployments require the virtualisation software to support network card bridging to allow the virtual machines to be "physically" connected to the laboratory infrastructure. Subjects that require a laboratory "search-by-name" facility (DNS and rDNS) such as Java sockets and C sockets programming are supported by the DDNS implementation as discussed in Sect. V-C.

VI. EXPERIENCE OF LABORATORY USE IN TEACHING

The general laboratory infrastructure has been in place, and actively used, for 10 years. The LIHV honeypot has been in place for 9 years. The honeypot was developed to support a cybersecurity module on an undergraduate networking programme and has been used successfully in the teaching of basic attack vectors such as port analysis, banner grabbing and service interaction (FTP and HTTP) both for taught modules and student projects.

The HILV honeypots were designed and built 6 years ago and have been used since 2012 (5 years) on undergraduate networking and cybersecurity programmes. They have been deployed in activities such as service redirection attacks, amplification attacks and man-in-the-middle scenarios. The use of Raspberry Pi boards in the HILV honeypots has allowed diverse subjects to be taught more easily due to their use of removable media to store the operating system. The setup time for laboratories and teaching sessions has been significantly reduced, in comparison with our previous approach of using small clusters of PC's with removable drives. This is reflected in the students' response to the configuration of the HILV honeypot, as discussed in section VI-D. There have been several hardware changes to the HILV honeypots over this time but the basic architecture has remained unchanged. The latest change involved an upgrade from Raspberry Pi 2 to

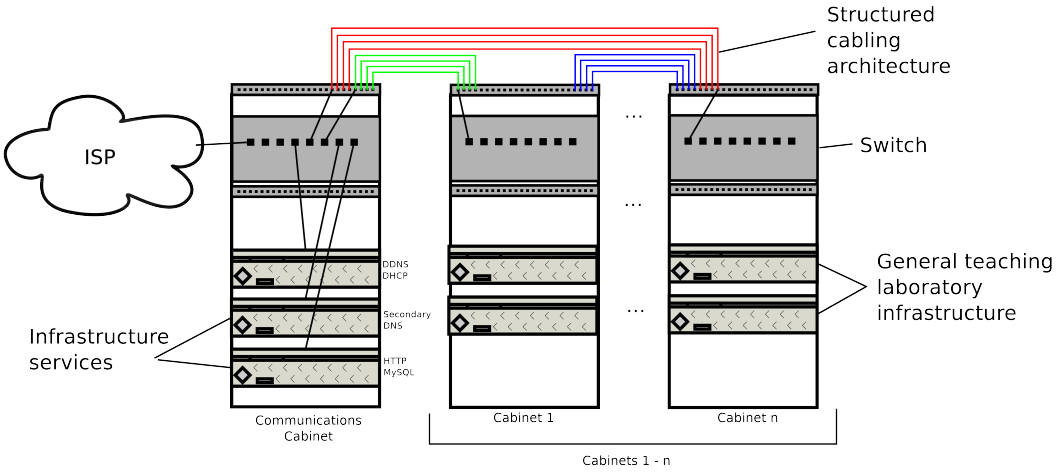

FIGURE 7. General networking laboratory overview

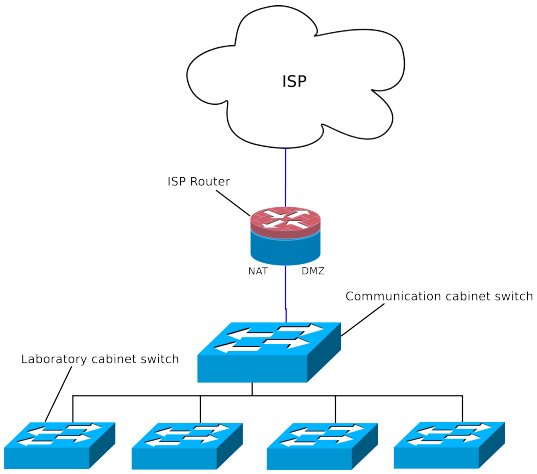

FIGURE 8. Logical view of internet access

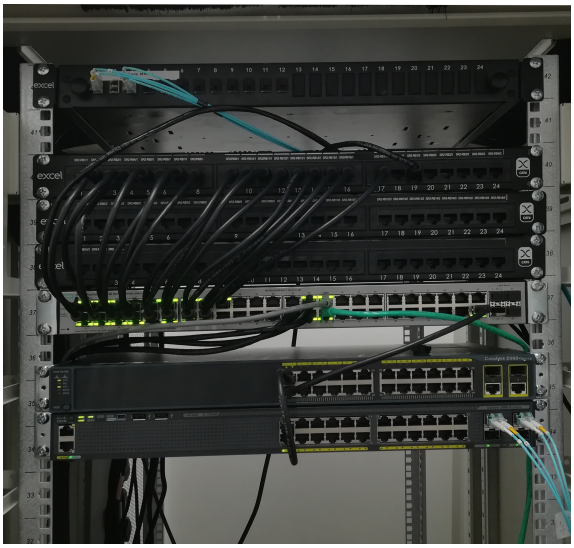

FIGURE 9. Communications cabinet

Raspberry Pi 3. The cost of layer 2 switches and their availability has also improved and they are now more affordable. As new honeypots are fabricated, TP-LINK switches are being used in preference to the more expensive HP switches. The TP-LINK switches do not provide a fully implemented monitor port; this requires the inclusion of a network tap as shown in Fig. 1.

A. SUPPORTED TEACHING

The current implementation of the laboratory supports >200 students. This includes undergraduates studying networking (≈40) and cybersecurity (≈150), and postgraduates studying networking (≈20). Each of our programmes is delivered in modules. A typical undergraduate programme runs around 10 modules concurrently e.g. networking technology (years 1, 2, 3 & 4 with MComp), security case projects (Year 2), sockets programming (year 3). Each module requires ≈3 hours contact per week and ≈3 hours of directed learning, which may require laboratory time. In addition, the laboratory supports many undergraduate and postgraduate cybersecurity projects (≈80), including cyber attack analysis and general cybersecurity research such as biometric-based, multi-factor authentication and IoT (Internet of Things) security projects.

B. SUPPORTED MODULES

All network engineering modules, across both undergraduate and postgraduate programmes, involving routing, switching, VLAN deployment, MPLS networking and IP telephony are successfully taught using the general networking laboratory infrastructure.

Network service deployment, using server operating systems (Windows and Linux), for both undergraduate and postgraduate programmes, is taught successfully in the environment using virtual machine technologies. The network service deployments include load-balanced HTTP, network

FIGURE 10. Communications cabinet servers

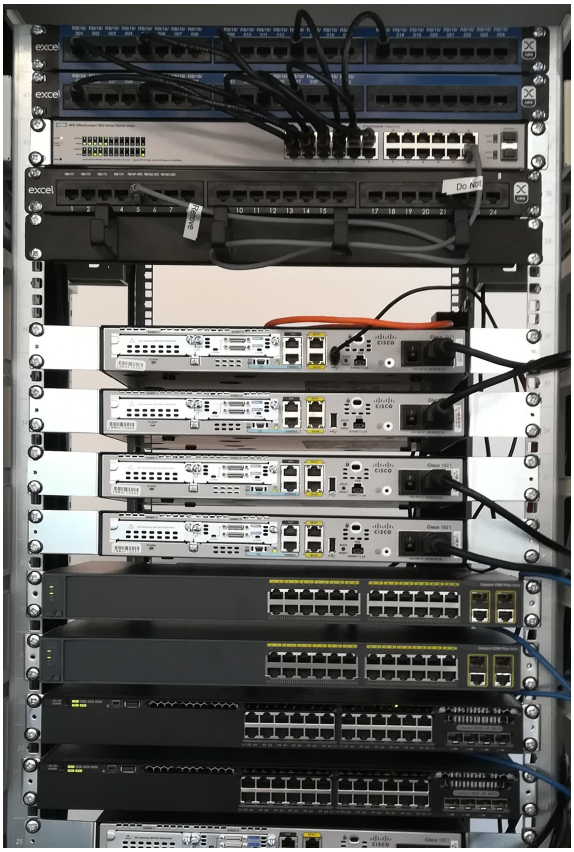

FIGURE 11. General networking laboratory cabinet

file system deployments (NFS and SMB), replication based MySQL services and large scale DNS deployments.

The HILV honeypot environment has allowed aspects of network-based infrastructure, specifically broadcast network services, to be taught as a practical implementation rather than only as a simulation, and has allowed students to develop complete network infrastructures integrated to the Internet.

Cybersecurity modules are predominantly taught using the HILV honeypots, particularly when looking at attack vectors that require packet spoofing or resource exhaustion through the volume of traffic generation.

The HILV honeypots have also allowed analysis of live attacks from the Internet without impacting on the local laboratory network. Activities such as port scanning are passed through directly to the HILV honeypot, without exposing the laboratory infrastructure. The use of multiple HILV honeypots has allowed profiles of subnet scanning and attacks to be analysed by students, providing them with a rich environment for experimentation and analysis.

C. SUPPORTED PROJECTS

The HILV honeypots have been available for several years now and students who move to the final year of the cybersecurity undergraduate programme tend to carry out research-based projects. Usually these involve the use of the basic HILV honeypot, but some projects add additional components to the honeypot such as firewalls (pfSense [51] or ipFire [52]) or wireless access points for the analysis of smartphone or tablet-based application attacks. A sample of recent projects is described below.

- *Multi-Tiered defence analysis of a simulated cyber attack.* This project involved configuring the HILV honeypots to support an IPFire (software based firewall technology), and investigating potential tunnelling techniques that could compromise a military grade network.
- *Development of a small IDS.* This project involved developing a libpcap–based application to run on one of the Raspberry Pi boards. The application monitored network traffic to identify a SYN flood attack using a simple window-based statistical analysis.
- *Attack on a secure IoT protocol.* This project involved developing a network of IoT devices for environment analysis (temperature and humidity), and identifying the encrypted payloads within the traffic (MQTT), which were then attacked using a block decryption technique.
- *Development of an IDS for a full subnet MitM attack.* This project involved developing a stateful IDS using libpcap to identify spoofed ARP packets.
- *Development of a DOS tool that attempts to prevent detection from an IDS.* This project involved developing a RAW sockets application that crafted packets to replicate valid traffic within the subnet environment.
- *Analysis of a DNS amplification attack.* This project involved configuring a vulnerable DNS environment, executing an attack, and analysing the bandwidth effect on the network.

D. STUDENT FEEDBACK

In academic year 2017/18, following the delivery of the security case project, we carried out an online, five-point, Likert-type scale survey to gauge student opinion about the effectiveness of the laboratory configurations. We canvassed a cohort of 48 students of whom 29 responded. The results of the survey are discussed briefly below.

- *Utility:* Students were asked how useful they found the HILV honeypot for the practical research component of the module. All students found the honeypot useful, with 89.66% rating the usefulness as very good or excellent. This result matched our expectations, as students usually report that they learn better when they have access to physical equipment.
- *Ease of use:* Students were asked how easy it was to configure the HILV honeypot. This question related to scenarios in which students were provided with a basic configuration and were required to extend and adapt it to their own requirements. 98% of students reported that it was not difficult to configure the honeypot, with the majority indicating that they found it easy or very easy. This ease of use can be attributed to the following factors.
  - The base configuration of the honeypot has a fixed configuration for its router and switch. This allows students to sign out any of the honeypots from the loans facility and use it immediately in a 'plug-and-play' fashion for many exercises.
  - The students created their own base servers from a set of images distributed from the laboratory NAS drive. Following each practical session, the students retained the SD cards for use at the next session. The SD cards provided a stateful configuration of their work.
  - The students were able to backup their entire project using a "dump" (dd) of the SD cards.
- *Previous experience and likely future use*: Students were asked if they had used a honeypot earlier in their studies. Most students (93%, 27/29) had not used a honeypot before their undergraduate studies and therefore had no preconception of what to expect. In contrast, when asked if they would consider using the HILV honeypot again, the majority (97%, 28/29) of the students indicated that they would use it for further studies.

More extensive and more careful studies are needed to quantify the pedagogical benefits of our approach but these preliminary results are an indication of a very positive student reaction to their experience in our laboratory.

VII. CONCLUSION AND FUTURE WORK

The development of the HILV honeypot environment, and its integration with the general purpose networks lab and the LIHV laboratory-based honeypot, has proved successful for both teaching and research. The cost of the HILV honeypot has been reduced to such an extent that, rather than the laboratory supporting a single (LIHV) honeypot, which has to be reconfigured between sessions, the laboratory can now support multiple honeypots that are highly configurable and portable.

As the HILV honeypots are small-scale and low-cost, the equipment is permanently configured for teaching purposes. Each HILV honeypot is capable of supporting four students at a time to work on research-based modules and allows practical cybersecurity modules to be delivered more effectively.

The integration of the honeypots into a laboratory environment that supports a wide-range of other technology-focused modules provides a cost-effective solution to the delivery of stimulating, practical cybersecurity teaching.

Student numbers have risen sharply for cybersecurity courses and the use of the honeypots has allowed additional levels of cybersecurity to be incorporated into existing network courses. This has had a strategic impact on the University since the B.C.S. (British Computing Society) added cybersecurity as a required part of its accreditation process.

Our aim is to increase the number of HILV honeypots to accommodate the increasing number of students. The low cost of the platform makes this an achievable goal. It is also envisaged that, using this technology, the department will be able to expand its cybersecurity research by preparing students for PhD studies in the subject area. It is intended to seek funding to develop a HIHV honeypot to support these PhD students. This honeypot will consist of several large scale servers along with commercial grade switches and routers and large-scale data capture facilities. Such a facility will provide an excellent environment for the study of cybersecurity at PhD level and has the potential to advance our understanding of the subject significantly.

REFERENCES

[1] R. T. Abler, D. Contis, J. B. Grizzard, and H. L. Owen, "Georgia tech information security center hands-on network security laboratory," IEEE Transactions on Education, vol. 49, no. 1, pp. 82–87, 2006.
[2] A. Lanoy and G. W. Romney, "A virtual honey net as a teaching resource," in 2006 7th International Conference on Information Technology Based Higher Education and Training, July 2006, pp. 666–669.
[3] T. Yang, K.-B. Yue, M. Liaw, G. Collins, J. T. Venkatraman, S. Achar, K. Sadasivam, and P. Chen, "Design of a distributed computer security lab," Journal of Computing Sciences in Colleges, vol. 20, pp. 332–346, 10 2004.
[4] D. Fraunholz, D. Krohmer, S. D. Anton, and H. D. Schotten, "Investigation of cyber crime conducted by abusing weak or default passwords with a medium interaction honeypot," in 2017 International Conference on Cyber Security And Protection Of Digital Services (Cyber Security), June 2017, pp. 1–7.
[5] M. L Bringer, C. Chelmecki, and H. Fujinoki, "A survey: Recent advances and future trends in honeypot research," in International Journal of Computer Network and Information Security, vol. 4, 09 2012.
[6] R. L. Clark, G. H. Flowers, P. Doolittle, K. Meehan, and R. W. Hendricks, "Work in progress - transitioning lab-in-a-box (liab) to the community college setting," in 2009 39th IEEE Frontiers in Education Conference, Oct 2009, pp. 1–6.
[7] T. Sochor, M. Zuzcak, and P. Bujok, "Analysis of attackers against windows emulating honeypots in various types of networks and regions," in 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), July 2016, pp. 863–868.
[8] A. Mairh, D. Barik, K. Verma, and D. Jena, "Honeypot in network security: a survey," in Proceedings of the 2011 international conference on communication, computing & security. ACM, 2011, pp. 600–605.

[9] R. McGrew, "Experiences with honeypot systems: Development, deployment, and analysis," in Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06), vol. 9, Jan 2006, pp. 220a–220a.
[10] I. Studnia, V. Nicomette, M. KaÃćniche, and E. Alata, "A distributed platform of high interaction honeypots and experimental results," in 2012 Tenth Annual International Conference on Privacy, Security and Trust, July 2012, pp. 229–230.
[11] G. K. Sadasivam and C. Hota, "Scalable honeypot architecture for identifying malicious network activities," in 2015 International Conference on Emerging Information Technology and Engineering Solutions, Feb 2015, pp. 27–31.
[12] M. Hibler, R. Ricci, L. Stoller, J. Duerig, S. Guruprasad, T. Stack, K. Webb, and J. Lepreau, "Large-scale virtualizationin the emulab network testbed," in Proceedings of USENIX Annual Technical Conference, Boston, MA, June 2008, 2008.
[13] M. Wannous and H. Nakano, "Nvlab, a networking virtual web-based laboratory that implements virtualization and virtual network computing technologies," IEEE Transactions on Learning Technologies, vol. 3, no. 2, pp. 129–138, April 2010.
[14] I. Marsa-Maestre, E. De La Hoz, J. M. Gimenez-Guzman, and M. A. Lopez-Carmona, "Design and evaluation of a learning environment to effectively provide network security skills," Computers & Education, vol. 69, pp. 225–236, 2013.
[15] R. T. Abler, D. Contis, J. B. Grizzard, and H. L. Owen, "Georgia tech information security center hands-on network security laboratory," IEEE Transactions on Education, vol. 49, no. 1, pp. 82–87, Feb 2006.
[16] K. Salah, M. Hammoud, and S. Zeadally, "Teaching cybersecurity using the cloud," vol. 8, no. 4, pp. 383–392, Oct 2015.
[17] Amazon Web Services, "AWS Customer Agreement," https://aws.amazon.com/agreement/, 2018, accessed: 2018-04-27.
[18] C. P. Lee, A. S. Uluagac, K. D. Fairbanks, and J. A. Copeland, "The design of netseclab: A small competition-based network security lab," IEEE Transactions on Education, vol. 54, no. 1, pp. 149–155, Feb 2011.
[19] Offensive Security, "Kali Linux | Penetration Testing and Ethical Hacking Distribution," https://www.kali.org/, 2017, accessed: 2017-11-19.
[20] Cyber Challenge Foundation, "Cyber Security Challenge UK," https://www.cybersecuritychallenge.org.uk/, 2018, accessed: 2018-04-29.
[21] L. Spitzner, Honeypots: tracking hackers. Addison-Wesley Reading, 2003, vol. 1.
[22] J. Duffany, "Design of a network security teaching and research lab," in Sixth International Latin American and Caribbean Conference for Engineering and Technology, June 2008.
[23] A. A. A. N. Albashir, "Detecting unknown vulnerabilities using honeynet," in 2015 First International Conference on Anti-Cybercrime (ICACC), Nov 2015, pp. 1–4.
[24] W. Fan, Z. Du, and D. Fernández, "Taxonomy of honeynet solutions," in 2015 SAI Intelligent Systems Conference (IntelliSys), Nov 2015, pp. 1002–1009.
[25] R. Kaur, E. S. Nagpal, and S. Chamotra, "Malicious traffic detection in a private organizational network using honeynet system," in 2015 Annual IEEE India Conference (INDICON), Dec 2015, pp. 1–6.
[26] A. Pandey and J. R. Saini, "ARP cache rectification for defending spoofing and poisoning attacks," in 3rd International Conference on Computing for Sustainable Global Development (INDIACom 2016), March 2016, pp. 3487–3492.
[27] D. R. Rupal, D. Satasiya, H. Kumar, and A. Agrawal, "Detection and prevention of ARP poisoning in dynamic IP configuration," in 2016 IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT), May 2016, pp. 1240–1244.
[28] Rapid 7, "Metasploit | Pentration testing software," https://www.metasploit.com/, 2017, accessed: 2017-11-19.
[29] R. Sankar, "Hydra-GTK Online Attack - Kali Linux Tools," http://kalilinuxtutorials.com/hydraonline/, 2015, accessed: 2017-11-24.
[30] E. Lewis, "RFC 5936 - DNS Zone Transfer Protocol (AXFR)," https://tools.ietf.org/html/rfc5936, 2010, accessed: 2017-11-24.
[31] Automattic Inc., "Blog Tool, Publishing Platform," https://wordpress.org, 2017, accessed: 2017-11-21.
[32] WPScan Team, "WPScan by the WPScan Team," https://wpscan.org, 2017, accessed: 2017-11-04.
[33] Raspberry Pi Foundation, "Products - Raspberry Pi," https://www.raspberrypi.org/products/, 2017, accessed: 2017-11-03.
[34] TP LINK, "TL-R470T+ Load Balanced Broadband Router," https://www.tp-link.com/uk/products/details/cat-4910_TL-R470T+.html, 2018, accessed: 2018-05-01.
[35] M. Bauer, "Paranoid penguin: Designing and using dmz networks to protect internet servers," Linux Journal, vol. 2001, no. 83es, p. 16, 2001.
[36] T. Sridhar, "Layer 2 and layer 3 switch evolution," in The Internet Protocol Journal, vol. 1, June 1998, pp. 38–43.
[37] Hewlett Packard, "Aruba 2530 8 Switch," https://www.hpe.com/us/en/product\-catalog/networking/networking\-switches/pip.overview.networking\-switches.5385011.html, 2017, accessed: 2017-10-30.
[38] TP-Link, "TL-SG2008," http://www.tp-link.com/us/products/details/cat-5070_TL-SG2008.html, 2017, accessed: 2017-10-30.
[39] TP-Link 8 Port Gigabit Easy Smart Switch, "TL-SG108E," https://www.tp-link.com/au/products/details/cat-41_TL-SG108E.html, 2018, accessed: 2018-05-10.
[40] R. Bejtlich, The practice of network security monitoring: understanding incident detection and response. No Starch Press, 2013.
[41] Hewlett Packard, "HPE ProLiant DL360 Gen9 Server," https://www.hpe.com/uk/en/product\-catalog/servers/proliant\-servers/pip.hpe\-proliant\-dl360\-gen9\-server.7252836.html, 2017, accessed: 2017-10-30.
[42] M. Bauer, Linux server security. O'Reilly Media, Inc., 2005.
[43] BT Plc, "Business Broadband," https://business.bt.com/products/broadband/, 2017, accessed: 2017-11-03.
[44] DrayTek Corp., "Vigor 2869 ADSL & VDSL," http://www.draytek.co.uk/products/business/vigor-2860, 2017, accessed: 2017-11-04.
[45] GNS3 Technologies Inc., "GNS3 | The software that empowers network professionals," https://www.gns3.com/, 2017, accessed: 2017-11-24.
[46] VMware, Inc, "VMWare - Official Site," https://www.vmware.com/, 2017, accessed: 2017-10-30.
[47] Oracle, Inc, "Oracle VM VirtualBox," https://www.virtualbox.org/, 2017, accessed: 2017-10-30.
[48] R. Aitchison, Pro DNS and BIND 10. Apress, 2011.
[49] R. E. Droms and T. Lemon, The DHCP handbook. Pearson Education, 2002.
[50] M. Stapp and B. Voltz, "RFC 4702 The Dynamic Host Configuration Protocol (DHCP) Client Fully Qualified Domain Name (FQDN) Option," https://tools.ietf.org/html/rfc4702, 2006, accessed: 2017-11-23.
[51] Rubicon Communications, LLC, "pfSense Firewall," https://www.pfsense.org/, 2018, accessed: 2018-05-01.
[52] IPFire team, "IPFire Project," https://www.ipfire.org/, 2018, accessed: 2018-05-01.

NEIL ELIOT is a Senior Lecturer at Northumbria University. Neil was awarded his Ph.D. from Northumbria in 2017, his Pg.C. in 1994, and his B.Sc. in 1989 all in Computing. His research areas include swarm theory and cybersecurity. Neil is a Certified Ethical Hacker (CEH) and coordinator for the EC Council Academic alliance at Northumbria. He is also responsible for the delivery of several network cybersecurity modules in the area of operating systems integration, smart home technologies and modules that focus on cybersecurity threats in large scale system deployments. Prior to this Neil worked in the chemical industry and the NHS.

DAVID KENDALL has been a Senior Lecturer in Computing at Northumbria University since 1989, except for a brief period as a Lecturer in Computer Science at Durham University (2001-02). Previously, he was a Research Associate at Newcastle University (1987-89), following experience as a Senior Software Engineer in industry (1983-86). He has an M.A. degree in Literae Humaniores from Oxford University, where he studied at New College. He received M.Sc. and Ph.D. degrees in Computing Science from Newcastle University. His research interests are in the areas of formal methods, embedded systems, distributed systems, security and privacy.

MICHAEL BROCKWAY has been a Senior Lecturer at Northumbria University for 17 years. His first degree is in pure and applied mathematics (1973-4), his Masters in mathematics (mathematical logic, category theory, 1975), and he has a PhD in computer science (formal methods for distributed embedded systems, 2010) from Northumbria University. Before that Michael worked in teaching and the computing industry.